

JACKSON LAWRENCE  
00000070612

LOUIS GABRIEL HERNANDES  
00000070250

# GENETIC ALGORITHM

OSCAR JIRO HARLISON  
00000072786

IGNATIUS STEVEN  
00000070642

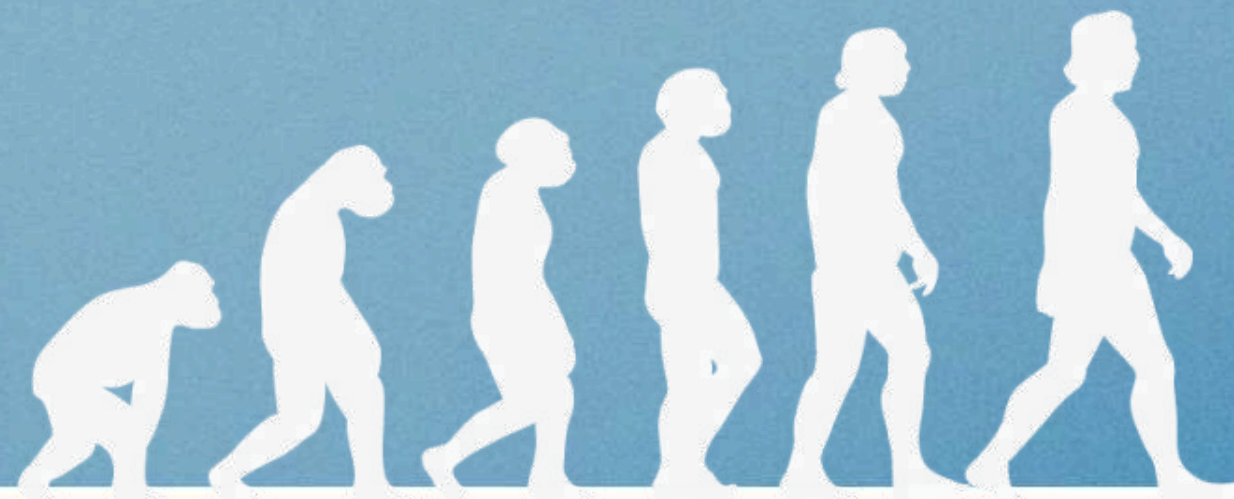


## DEFINITION

- **Genetic Algorithm (GA)** adalah sebuah algoritma metaheuristik dari cabang Evolutionary Algorithms (EA) untuk memecahkan masalah optimasi *constrained* maupun *unconstrained*.
- Dikenalkan pertama kali oleh **John Holland** pada **1975**, GA terinspirasi dari konsep *natural selection* dan evolusi biologi.

## NATURAL SELECTION

- Dalam teori evolusi biologi, semua makhluk hidup berevolusi untuk beradaptasi dengan lingkungan dan meningkatkan kemampuan bertahan hidup dan berduplikasi; proses ini disebut juga *natural selection*
- Karena seleksi natural ini, keturunan pun pada umumnya lebih berkualitas dari induknya
- Setiap generasi keturunan baru membentuk populasi unik karena mutasi yang terjadi dan gabungan gen dari induk-induknya yang terus berevolusi (*crossover*)





## WHY USE G.A.?

- Mampu **secara efisien melakukan pencarian di dataset yang besar dan kompleks**
- **Fleksibel** karena dapat digunakan untuk banyak jenis masalah seperti pencarian atau optimasi
- Karena adanya mutasi, GA **mampu menghindari local optima** dengan baik
- **Mudah dimengerti tanpa perlunya analisis mendalam** apabila representasi solusi individu tepat
- Mampu dijalankan secara **paralel**

## APPLICATIONS

- **DNA Analysis**
- **Vehicle Routing**
- **Traveling Salesman Problem**
- **Optimizing Functions**
- **Knapsack Problem**
- **Scheduling Problem**

## HOWEVER....

- Layaknya semua algoritma metaheuristik, jawaban **tidak deterministik**, sehingga tidak selalu sama atau **tidak menjamin jawaban ter-optimal**; untuk hal yang sama, tahap mutasi dapat seperti sebuah double-edged sword
- Berpotensi membutuhkan banyak generasi untuk mencapai solusi optimal karena **konvergensinya yang juga berpotensi lama** akibat sifat probabilitiknya
- **Memerlukan penyesuaian per masalah yang berbeda**, mulai dari encoding populasi, definisi fitness function, hingga parameter seperti ukuran populasi, tingkat mutasi, dan tingkat crossover yang tentunya menentukan keefektifan algoritma



## HOW EVOLUTION REALLY RELATES TO GENETIC ALGORITHM

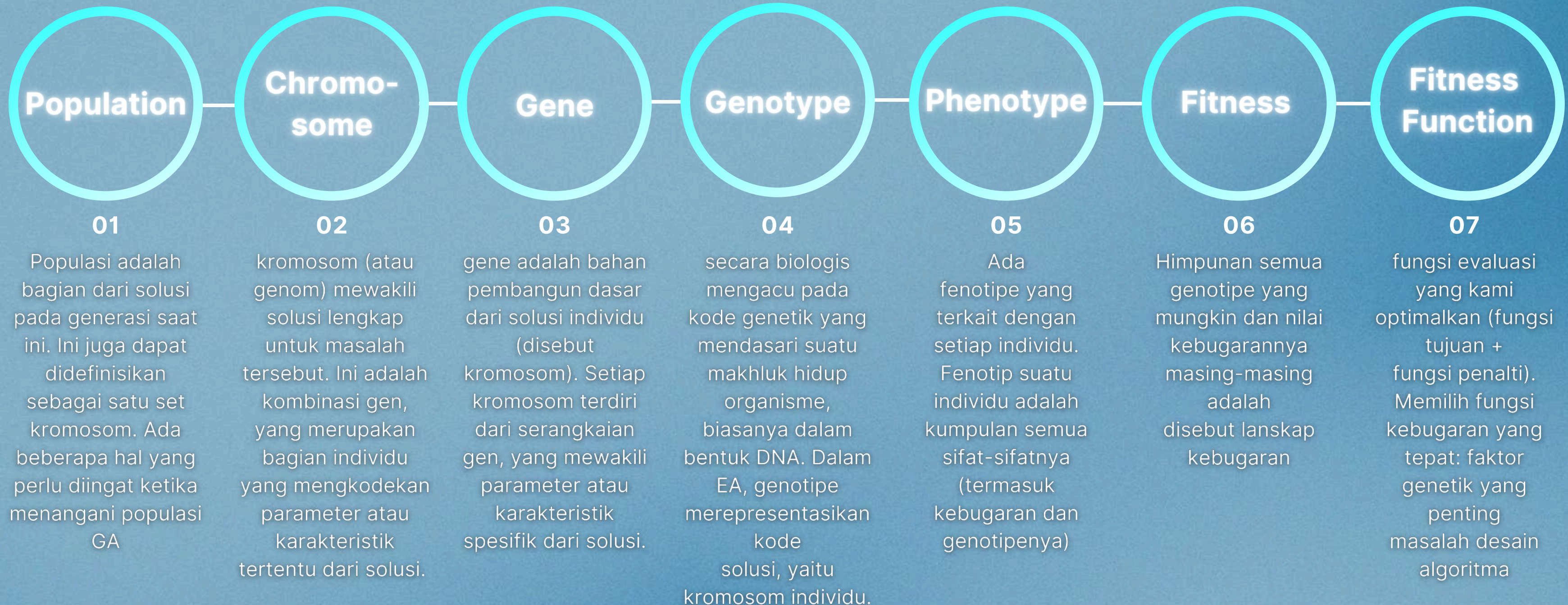
- Setelah beberapa generasi, populasi telah berevolusi menjadi makhluk hidup yang jauh lebih berkualitas dan *fit*
- Konsep tersebut yang membuat GA dapat dilakukan
- Mulai dari suatu kumpulan solusi yang acak (populasi), solusi-solusi generasi demi generasi akan berevolusi hingga solusi yang optimal tercapai

## OVERALL CONCEPT





# TERMINOLOGY / BASIC CONCEPT





# SELECTION

## Roulette Wheel Selection

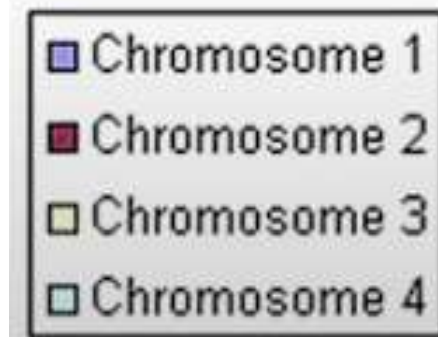
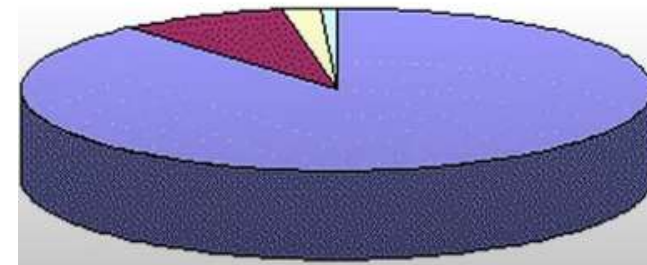
- Parents diselect menurut fitnessnya
- Chromosom yang lebih baik, lebih berkemungkinan besar untuk diselect
- Perhitungan dengan konsep Probabilitas

Individu	Fitness	
A	10	$P(A) = \frac{10}{210} \approx 0.048$ (4.8%)
B	30	$P(B) = \frac{30}{210} \approx 0.143$ (14.3%)
C	20	$P(C) = \frac{20}{210} \approx 0.095$ (9.5%)
D	40	$P(D) = \frac{40}{210} \approx 0.190$ (19.0%)
E	60	$P(E) = \frac{60}{210} \approx 0.286$ (28.6%)
F	50	$P(F) = \frac{50}{210} \approx 0.238$ (23.8%)

$\Sigma \text{Fitness} = 210$   $\therefore$  Lebih diselect  $\rightarrow$  E lalu F, dst.

## Ranking Selection

- Chromosom disort berdasarkan nilai fitnessnya pada individu yang paling besar ke yang paling kecil



$\therefore$  Lebih diselect  $\rightarrow$  Chromosome 1 lalu 2, dst.

## Truncation Selection

- Memilih individu dengan fitness tertinggi 50% dari seluruh chromosom

Individu	Fitness
A	10
B	30
C	20
D	40
E	60
F	50

$\therefore$  Lebih diselect  $\rightarrow$  Individu E, F, D karena berada pada 50% teratas

## Tournament Selection

- Beberapa individu secara acak dan membandingkan fitness mereka, lalu memilih yang terbaik

Contoh 1:

1. Acak pilih tiga individu: B, D, F.
2. Fitness mereka adalah: B (30), D (40), F (50).
3. Individu F dipilih karena memiliki fitness tertinggi.

Contoh 2:

1. Acak pilih tiga individu: A, C, E.
2. Fitness mereka adalah: A (10), C (20), E (60).
3. Individu E dipilih karena memiliki fitness tertinggi.



# CROSSOVER

## Single Point Crossover

Chromosome1	11011   00100110110
Chromosome2	11011   11000011110
Offspring1	11011   11000011110
Offspring2	11011   00100110110

## Double Point Crossover

Chromosome1	11011   00100   110110
Chromosome2	10101   11000   011110
Offspring1	11011   11000   110110
Offspring2	10101   00100   011110

## Uniform Crossover

Parent :	000000000000000000000000
	111111111111111111111111
Children :	100011010100100111101
	011100101011011000010

## NOTES :

- Sebelum tahap selection, setiap permasalahan biasanya diencode dengan biner yaitu **binary encoding** (0 atau 1)
- Ada juga 2 tipe encoding lainnya seperti berikut ini.

**Permutation Encoding** : Tiap kromosom adalah string penomoran dalam barisan

Chromosome A	153264798
Chromosome B	856723149

**Value Encoding** : Tiap kromosom adalah string dari suatu nilai (Bilangan, bilangan rill, char)

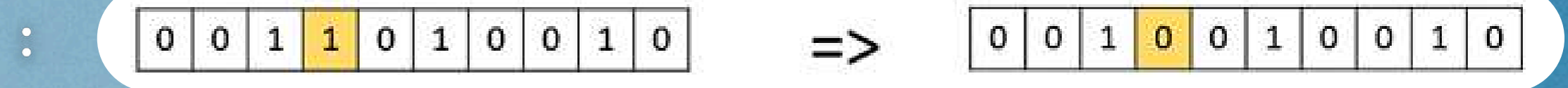
Chromosome A	3	7	2	6	11
Chromosome B	1.2321	5.4367	2.4243	0.4563	3.5432
Chromosome C	(back)	(back)	(right)	(forward)	(left)



# MUTATION

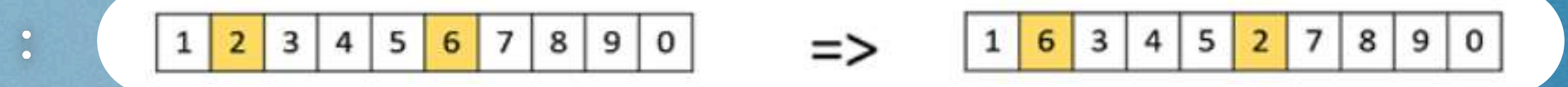
## Bit Flip Mutation

- Menukar 0 dengan 1, dan sebaliknya



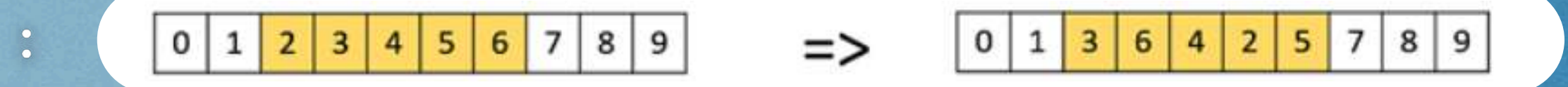
## Swap Mutation

- Menukar posisi individu dalam chromosom



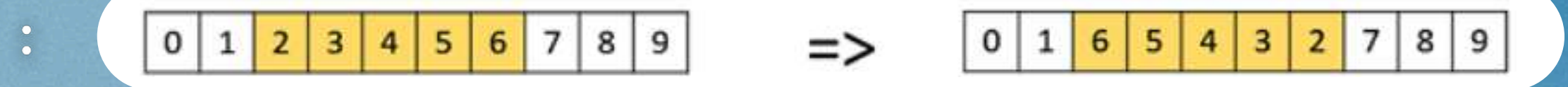
## Scramble Mutation

- Mengacak posisinya dalam chromosom



## Inversion Mutation

- Dibalik urutannya posisi dalam chromosom



Contoh

Individu A: Fitness = 25

Individu B: Fitness = 20

Individu C: Fitness = 18

Individu D: Fitness = 15

- A memiliki fitness tertinggi.
- Untuk generasi selanjutnya bebas dengan dipertahankan 1 atau 2 individu terbaik dari generasi awal, dan crossover/mutasi akan diterapkan pada individu lainnya untuk mengisi sisa populasi

## NOTES :

- Setelah proses Selection  $\rightarrow$  Crossover  $\rightarrow$  Mutation selesai dapat dilakukan proses **Etilism**.
- **Etilism** adalah strategi yang menjaga beberapa individu dengan fitness tertinggi dari generasi sebelumnya (generation 0) dan memastikan mereka tetap ada di generation 1.



# PSEUDOCODE

## Genetic Algorithm

```
function genetic_algorithm(population_size, num_generations, crossover_rate, mutation_rate):  
    population = initialize_population(population_size)  
  
    for generation in range(num_generations):  
        fitness_scores = evaluate_fitness(population)  
        selected_parents = select_parents(population, fitness_scores)  
  
        new_population = []  
        for i in range(population_size):  
            parent1, parent2 = select_parents(population, fitness_scores)  
            child = crossover(parent1, parent2, crossover_rate)  
            child = mutate(child, mutation_rate)  
            new_population.append(child)  
  
        population = new_population  
  
    best_individual = find_best_individual(population, fitness_scores)  
    return best_individual
```

## TIME COMPLEXITY

$$O(N \times P \times (P + C))$$

## SPACE COMPLEXITY

$$O(P \times C)$$

dimana :

- N = Num\_generations
- P = Population\_size
- C = Chromosome\_length

<https://github.com/KriezAlf/ExSys/tree/main/Genetic%20Algorithm>



# KNAPSACK PROBLEM

## POPULATION

0, 0, 0, 0 = 0KG

0, 0, 0, 1 = 2KG

0, 0, 1, 0 = 7KG

0, 0, 1, 1 = 9KG

0, 1, 0, 0 = 4KG

0, 1, 0, 1 = 6KG

0, 1, 1, 0 = 11KG

0, 1, 1, 1 = 13KG

1, 0, 0, 0 = 5KG

1, 0, 0, 1 = 7KG

1, 0, 1, 0 = 12KG

1, 0, 1, 1 = 14KG

1, 1, 0, 0 = 9KG

1, 1, 0, 1 = 11KG

1, 1, 1, 0 = 0KG

1, 1, 1, 1 = 0KG



A  
5KG



B  
4KG



C  
7KG



D  
2KG



Bag Limit  
15KG

### Selection

0, 1, 0, 1  
0, 0, 1, 1



### Crossover

0, 1, 0, 1  
0, 0, 1, 1

=

0, 1, 1, 1  
0, 0, 0, 1



### Mutation

0, 1, 1, 0  
0, 0, 0, 1

Red is mutated



Ini diulangkan  
sebanyak iterasi  
yang ditentukan  
untuk  
mendapatkan  
children dengan  
fitness value  
yang maksimal



OPTIMIZING A FUNCTION

$$f(x) = 0.2x^2 + \frac{450}{x}$$

$$0.5 \leq x \leq 25.5$$

Karena actual count sama, berarti crossover bebas

1 >< 4

2 >< 3

4.2 >< 23.5

10.1 >< 16.4

=

00101010 >< 11101011

01100101 >< 10100100

Single Crossover

0010 | 1010 >< 1110 | 1011

0110 | 0101 >< 1010 | 0100

=

0010 | 1011 >< 1110 | 1010

0110 | 0100 >< 1010 | 0101

Swap Mutation

10101011 >< 11111010

11100100 >< 10110101

=

17.1 >< 25.0

23.2 >< 18.5

∴ **Generation 1 = [17.1, 25.0, 23.2, 18.5]**  
(Jika memakai konsep Etilism maka ambil top 2 best dari generation 0 dan 1)

x value Generation 0						
i	Initial	Initial Pop. (Binary)	f(x) value	Prob. Count	Exp. Count	Actual Count
1	4.2	00101010	110.67	0.256	1.15	1
2	10.1	01100101	64.95	0.168	0.67	1
3	16.4	10100100	81.23	0.210	0.84	1
4	23.5	11101011	129.59	0.335	1.34	1

11101011

128+64+32+8+2+1 = 235

(Anggap 23.5)

$= 0.2 \cdot (23.5)^2 + 450 / 23.5$

$= 110.45 + 19.14$

$= 129.59$

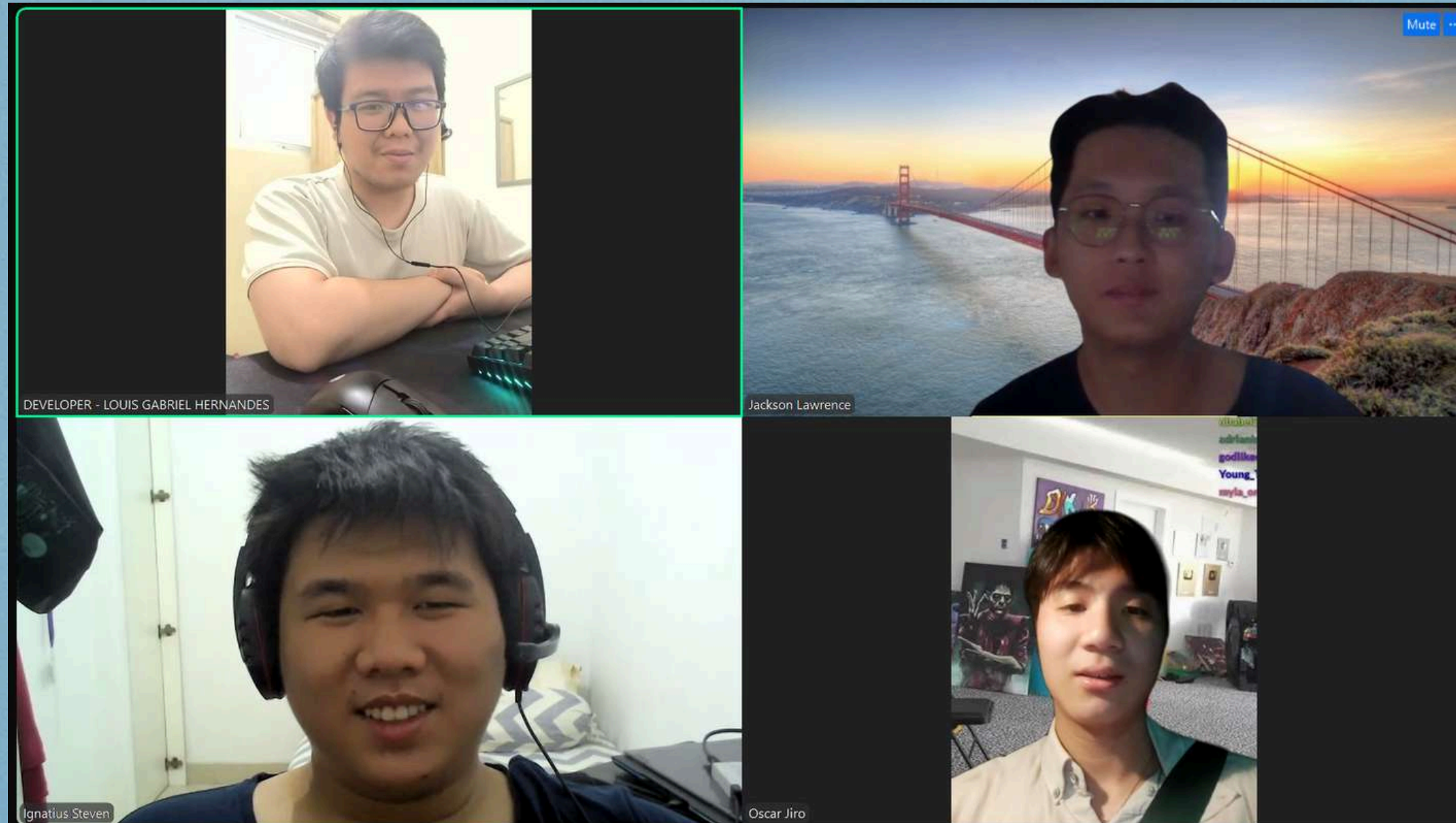
$= 129.59 / 386.44$

$= 0.335$

$= 129.59 / 96.61$

$= 1.34$



**Links:**

- <https://chatgpt.com/share/5a3a91a0-a3fe-4e7a-a536-8eea7201d32c>
- <https://rosyid.lecturer.pens.ac.id/materi%20AI/Minggu10%20-%20Algoritma%20Genetika.pdf>
- <https://chatgpt.com/share/0536a70c-147f-4321-ba7e-8812ceb81b1f>
- <https://chatgpt.com/share/e115f8d4-f6d5-45de-8b2a-c106b2f72548>
- <https://g.co/gemini/share/bcd1429fd022>
- <https://medium.com/@byanalytixlabs/a-complete-guide-to-genetic-algorithm-advantages-limitations-more-738e87427dbb>
- <https://becominghuman.ai/understanding-genetic-algorithms-a-use-case-in-organizational-field-2087c30fb61e>
- [https://www.tutorialspoint.com/genetic\\_algorithms/genetic\\_algorithms\\_population.htm](https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_population.htm)
- <https://www.geeksforgeeks.org/crossover-in-genetic-algorithm/>
- [https://www.tutorialspoint.com/genetic\\_algorithms/genetic\\_algorithms\\_mutation.htm](https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_mutation.htm)

**Paper:**

- Materi PPT Pertemuan 3 Expert System Universitas Multimedia Nusantara
- [https://datajobstest.com/data-science-repo/Genetic-Algorithm-Guide-\[Tom-Mathew\].pdf](https://datajobstest.com/data-science-repo/Genetic-Algorithm-Guide-[Tom-Mathew].pdf)

**Video:**

- [https://youtu.be/emcdnQzES7E?si=Ucc5VOj8A\\_-7NBto](https://youtu.be/emcdnQzES7E?si=Ucc5VOj8A_-7NBto)
- <https://youtu.be/1CIIIk6saak?si=Vf8IrrLEkZ7jf5Qk>
- <https://www.youtube.com/watch?v=ul5viW4r5ic&t=2214s>
- <https://www.youtube.com/watch?v=uQj5UNhCPuo&pp=ygURZ2VuZXRpYyBhbGdvcmI0aG0>
- <https://www.youtube.com/watch?v=XP2sFzp2Rig&pp=ygURZ2VuZXRpYyBhbGdvcmI0aG0>
- <https://www.youtube.com/watch?v=JgqBM7JG9ew>
- <https://www.youtube.com/watch?v=qikW1qX97qA&pp=ygURZ2VuZXRpYyBhbGdvcmI0aG0>
- <https://www.youtube.com/watch?v=-kpcAa-qKwY&pp=ygURZ2VuZXRpYyBhbGdvcmI0aG0>